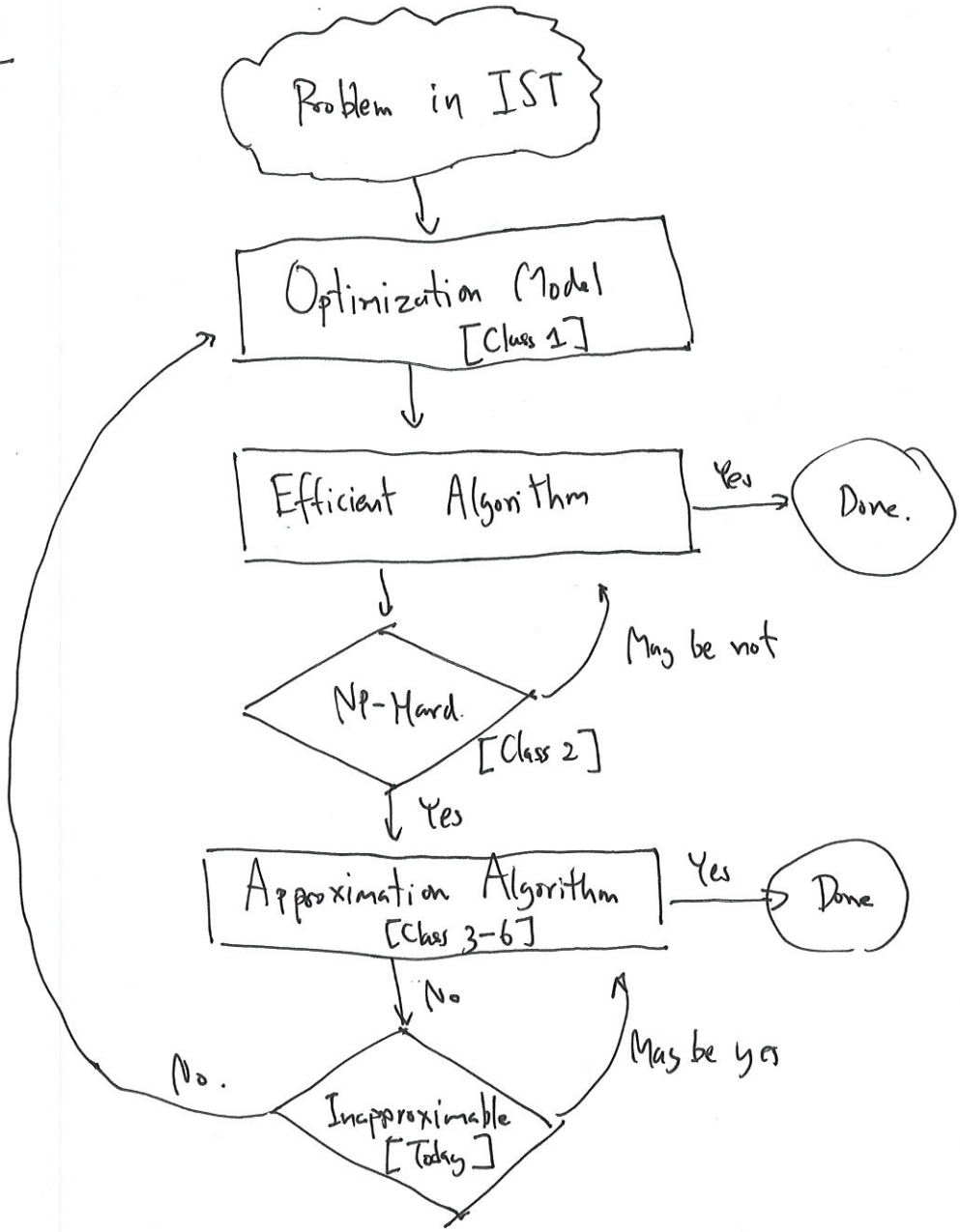
Approximation and Online Algorithms with Applications

#6

Until Now



Question

- Is it possible to find an approx. alg. for our model?
- ~~Is~~ Can we still improve our ~~optimization~~ approx. ratio?

k-center (NP-Hard)

Input: Point position $p_1, \dots, p_n$, # selections k

Output: $S \subseteq \{1, \dots, n\}$

Constraint: $|S| = k$

Objective Function: d_{ij} : distance between p_i and p_j

$D_i^{(S)} = \min_{j \in S} d_{ij}$: distance for i to be at a ward office.

$D_S = \max_i D_i^{(S)}$: longest commute time.

maximize D_S ||

Reformulate the problem (NP-Hard)

Input and Output : Same

Constraint : $|S| = k$, and $D_S = OPT$.

Objective Functive : None.

{}

Relax the problem (2-approx k-center)

Constraint : $|S| = k$, and $D_S \leq 2 \cdot OPT$

(Solved by approx. alg. on last week).

How about ? (1.999-approx. k-center)

Constraint: $|S| = k$, and $D_S \leq 1.999 \cdot OPT$.

Theorem 1.999-approx. k-center problem is NP-hard.

Corollary α-approx. k-center problem is NP-hard when $\alpha \leq 1.999$

Proof

```
Set 1.5-approx. k-center (point[] p, int k)
    // return |S|=k, and D_S ≤ 1.5·OPT

Set 1.999-approx k-center (point[] p, int k){
    // should return |S|=k, and D_S ≤ 1.999·OPT
    return 1.5-approx. k-center (p, k)
}
```

∎

Dominating Set Problem (Decision Version) [NP-Hard]

Input: A social network (V, E), positive integer k.

Output: Yes / No

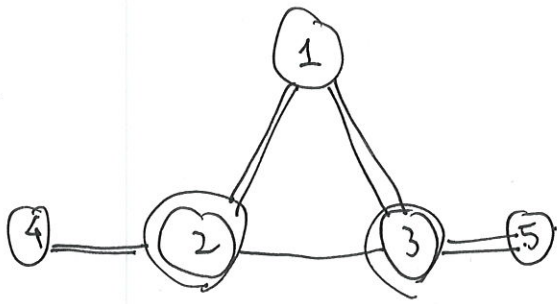
Constraint: Yes, when there is $S \subseteq V$ such that

1. $|S| = k$
2. All persons are in S or can receive a message from S.

No, otherwise.

Objective Function None.

Example



$k=2 \rightarrow$ yes

(2) and (3)

$k=1$ = no.

Proof of theorem [Hsu and Newhouser, 1979]

```
Set 1.999-approx k-center (point[] p, int k);
    // provided in a library.
bool DominatingSet (Set V, Set E, int k) {
    Suppose that V = {1, ..., n}
    p[i] denotes person i
    d(p[i], p[j]) = { 0  if i = j
                    { 1  if {i,j} ∈ E
                    { 2  otherwise.
    S ← 1.999-approx k-center (p, k)
    If all persons are in S or can receive a message from S :
        return true.
    Else
        return false. ) ? why.
}
```

we have a set S that $|S| = k$ and condition 2 is satisfied.

All distances are 1 and 2.

Commute time are 1 or 2. $\rightarrow$ OPT, SOL $\in \{1, 2\}$

OPT = 2 $\longrightarrow$ SOL = 2 (SOL cannot smaller than OPT)

OPT = 1 $\longrightarrow$ SOL = 1 ($SOL \leq 1.999 \cdot OPT$)

opt = 0 $\longrightarrow$ SOL = 0 ($SOL \leq 1.999$, $SOL \leq 1.999 \cdot OPT$, $SOL \leq 0$)

$$d_{ij} = \begin{cases} 0 & \text{if } i=j \\ 1 & \text{if } \{i,j\} \in E \\ 2 & \text{otherwise} \end{cases}$$

Commute time for i $\quad D_i^{(S)} = \min_{j \in S} d_{ij} = \begin{cases} 0 & \text{if } i \in S. \\ 1 & \text{if } i \notin S \text{ and there is } j \in S \text{ such that } \{i,j\} \in E \\ 2. & \text{if } i \notin S, \text{ and no } j \in S \text{ such that } \{i,j\} \in E \end{cases}$

$$D_S = \max_i D_i^{(S)} \quad D_S = \begin{cases} 0 & \text{if } i \in S \text{ for all } i \\ & \quad \text{all persons} \quad \hookrightarrow \text{impossible because } |S| = k < n. \\ 1 & \text{if, for all } i, \ i \in S \\ & \text{or there is } j \in S \text{ such that } \{i,j\} \in E. \\ & \quad \hookrightarrow \text{can receive a message from } S. \\ & \quad \hookrightarrow \text{condition 2.} \\ 2 & \text{otherwise.} \end{cases}$$

- If there is S such that
  1) $|S| = k$
  2) all persons are in S or can receive a message from S.

  $\Downarrow$ $\quad D_S = 1$

  1.9999-approx. k-center returns that S.

- If 1.999-approx. k-center returns S such that $D_S = 2$,

  then no S with the property ∎

Applications Allocation in Volunteer Cloud

3(-176772
+1.67

[Jiang, Wang, Cérin, Gianess, Ngoko, IPSPS 2014 (workshop)]

Hangzhou Dianzi Paris 13.

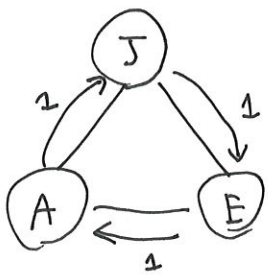
- Volunteer Cloud — Does not open 24 hours

Ex

Japan	8:00 – 17:00	JST
Europe	16:00 – 1:00	JST
America	0:00 – 9:00	JST

- We want our applications available for any time.
- There is a cost for migrating an application.

$C_{jii'}$: cost for migrating applicatim j from server i to i'



$C_{1JE} + C_{1EA} + C_{1AJ}$.

- Each cloud can host at most k applications.

Optimization Model

Input : t : # time slots $T = \{1, \dots, T\}$ Maximum Capacity : k.

n : # cloud servers $V = \{1, \dots, n\}$

$S_1, S_2, \dots, S_n \subseteq T$: Available time slots for Server i

$c_{jii'}$ ~~for all $j \in T$~~ .

m : # applications $M = \{1, \dots, m\}$

$c_{jii'}$ for all $j \in M$ and $i, i' \in V$: Migration cost

Output : $p_j^{(t)} \in \{1, \dots, n\}$: position of App j at time t

Constraint : For t and i, $\left| \{ j, p_j^{(t)} = i \} \right| \le k$

Set of Apps in cloud server i at time t

$p_j^{(t)} \in S_t$

Objective function Minimize $C_1 p_1^{(1)} p_1^{(2)} + C_1 p_1^{(2)} p_1^{(3)} + \dots + C_1 p_1^{(n-1)} p_1^{(n)} +$

$C_2 p_2^{(1)} p_2^{(2)} + C_2 p_2^{(2)} p_2^{(3)} + \dots + C_2 p_2^{(n-1)} p_2^{(n)} +$

$\vdots$

$C_m p_m^{(1)} p_m^{(2)} + C_m p_m^{(2)} p_m^{(3)} + \dots + C_m p_m^{(n-1)} + p_m^{(n)}$

position of app at time t must be in an available server list S_t

$$= \sum_{j=1}^{m} \sum_{t=1}^{T-1} C_j p_j^{(t)} p_j^{(t+1)}$$

Minimize $\sum_{j=1}^{m} \sum_{t=1}^{T-1} C_j p_j^{(t)} p_j^{(t+1)}$

3D Assignment Problem

- We have n tasks, n workers, and n place of work.
- Each worker can work at one task, and, each place can accomodate one worker.
- C_{ijk} is a cost for assigning worker i to task j and place k.
- We want to minimize a total cost.

Optimization Problem

Input: n : # workers, tasks, places. → same as input when $m=n$.

c_{ijk} : cost

Output: $p_i^{(1)} \in \{1, \dots, n\}$: task of worker i

$p_i^{(2)} \in \{1, \dots, n\}$: place for worker i → same as output of Application Allocation problem when $\tau = 3, k=2$, $S_1 = S_2 = \{1, \dots, n\}$

Constraint:

$p_i^{(1)} \neq p_j^{(1)}$ when $i \neq j$

$p_i^{(2)} \neq p_j^{(2)}$ when $i \neq j$

→ same as constraints of Application Allocations problem when $k=1$

Objective Function: Minimize $C_{1p_1^{(1)}p_1^{(2)}} + C_{2p_2^{(1)}p_2^{(2)}} + \dots + C_{np_n^{(1)}p_n^{(2)}}$

Minimize $\sum_{i=1}^{n} C_{ip_i^{(1)}p_i^{(2)}}$ → same as objective function of Application Allocations problem when $\tau = 2$

Theorem [Kann, Inform. Process. Lett. 1991]

There is $\alpha > 1$ such that no α-approximation algorithm for 3-dimensional assignment problem.

(3D-assignment is APX-hard problem)

Constraint $SOL_{AA} = \sum_{i=1}^{n} C_{ip_i^{(1)}p_i^{(2)}} \leq \alpha \cdot OPT_{AA}$ → NP-Hard.

Objective Function None

Theorem There is $\alpha > 1$ such that no α-approximation algorithm for application assignment problem.

Additional Constraint	$SOL_{3A} \leq \alpha \cdot OPT_{3A}$	NP-Hard
Objective Function	None	

Proof

```
int[][] ApplicationAssignment(int t, int n, int m, Set[] S, int[][][] c);
// in library    SOL_AA ≤ α·OPT_AA
int[][] 3DAssignment(int n, int[][][] c) {
    return ApplicationAssignment(2, n, n, {{1,...,n},{1,...,n}}, c);
}
```

$$SOL_{3A} = SOL_{AA} \leq \alpha \cdot OPT_{AA} = \alpha \cdot OPT_{3A}$$

$$SOL_{3A} \leq \alpha \cdot OPT_{3A}$$

∎